EXPRESS MAIL LABEL NO.: EL953523587US
PATENT APPLICATION
DOCKET NO.: CTX-079
(1545/133)

SECURE TRAVERSAL OF NETWORK COMPONENTS

Field Of The Invention

[0001] The present invention relates generally to traversing network components and, more specifically, to providing secure, authenticated traversal of arbitrary network components using next-hop routing and per-hop tickets.

Background of the Invention

[0002] Referring to FIG. 1, a computer system 100 known to the prior art typically includes a client computer 110, a content server proxy 115, and a content server 120. The client computer 110 is typically a personal computer that can download information from the content server 120 over a network 130, such as the Internet or World Wide Web. The content server proxy 115 is typically a security gateway, such as a router, through which messages to and from the content server 120 pass. The content server 120 hosts one or more application programs that can be accessed by the client 110.

[0003] The client 110 is typically in communication with the content server proxy 115 over a client-proxy communication channel 135. The content server proxy 115 is typically in communication with the content server 120 over a proxy-server communication channel 145. The computer system 100 also typically includes firewalls 150, 160 to prohibit unauthorized communication to/from the content server 120.

[0004] The client 110 typically gains access to the content server 120 after passing through the firewall 150 of the content server proxy 115 and the firewall 160 of the content

server 120. Thus, if an unauthorized user bypasses the content server proxy 115 and the firewall 160 (that is, if an unauthorized user is able to connect to the content server 120 without first accessing the content server proxy 115) the unauthorized user can typically access the content server 120 without encountering additional security. Further, a malicious user breaching firewall 150 typically has unrestrained access to the content server proxy 115 and, in many cases, to content server 120.

[0005] Therefore, there is a need to increase the protection of a content server 120 from an unauthorized user. There is also a need to enforce network routing requiring the client 110 to pass through one or more additional security measures before gaining access to the content server 120.

Summary of the Invention

[0006] The present invention relates to a method and system for authenticating a client to a content server. In one aspect, the method includes the step of generating a ticket, by a ticket authority, associated with the client. The ticket comprises a first ticket and a second ticket. The method also includes the steps of transmitting the first ticket to the client and the client using the first ticket to establish a communication session with a content server proxy. The method also includes the steps of transmitting the second ticket to the content server proxy and the content server proxy using the second ticket to establish a communication session with the content server.

[0007] In one embodiment, the client is authenticated to a web server before the ticket authority generates the ticket associated with the client. The method may also include the step of transmitting the first ticket to a web server and the web server transmitting the first ticket to the client. In another embodiment, the ticket authority transmits a disabled second

ticket with the first ticket to the client. The ticket authority can also transmit the address of a content server with the transmission of the second ticket to the content server proxy.

[0008] In another aspect, the system includes a client, a ticket authority, a content server, and a content server proxy. The content server proxy communicates with the client, the ticket authority, and the content server. The ticket authority generates a ticket associated with the client. The ticket comprises a first ticket and a second ticket. The first ticket is transmitted to the client and used to establish a first communication session with the content server proxy. The second ticket is transmitted to the content server proxy and used to establish a second communication session with the content server.

[0009] In one embodiment, the client is authenticated to a web server. The ticket authority can also transmit the second ticket to the web server and the web server transmits the second ticket to the content server for validation. In one embodiment, the content server proxy is a secure socket layer relay.

Brief Description of the Drawings

[0010] The advantages of the invention described above, together with further advantages, may be better understood by referring to the following description taken in conjunction with the accompanying drawings. In the drawings, like reference characters generally refer to the same parts throughout the different views. Also, the drawings are not necessarily to scale, emphasis instead generally being placed upon illustrating the principles of the invention.

[0011] FIG. 1 is a block diagram of an embodiment of a prior art communications system.

[0012] FIG. 2A is a block diagram of an embodiment of a communications system constructed in accordance with the invention.

[0013] FIG. 2B is a block diagram of another embodiment of a communications system constructed in accordance with the invention.

[0014] FIG. 3 is a flow diagram illustrating an embodiment of the operation of the communications system of FIG. 2A in accordance with the invention.

[0015] FIG. 4A is a block diagram of another embodiment of a communications system constructed in accordance with the invention.

[0016] FIG. 4B is a flow diagram illustrating an embodiment of the operation of the communications system of FIG. 4A in accordance with the invention.

Detailed Description of the Invention

[0015] FIG. 2A shows a block diagram of an embodiment of a communications system 205 for secure delivery of content. The communications system 205 includes the client 110, the content server proxy 115, the content server 120, a web server 220, and a ticket authority 225. The communications system 205 also includes the two firewalls 150, 160 which prohibit unauthorized communications to/from the content server 120. The network between the firewalls 150, 160 is often referred to as a "demilitarized zone," (DMZ) 230. In one embodiment, the DMZ 230 includes the content server proxy 115 and the web server 220.

[0016] The client 110 can be any personal computer (e.g., based on a microprocessor from the x86, 680x0, PowerPC, PA-RISC, MIPS families), smart or dumb terminal, network computer, wireless device, information appliance, workstation, minicomputer, mainframe computer or other computing device that has a graphical user interface. Operating systems supported by the client 110 can include any member of the WINDOWS family of operating systems from Microsoft Corporation of Redmond, Washington, Macintosh operating system,

JavaOS, and various varieties of Unix (e.g., Solaris, SunOS, Linux, HP-UX, A/IX, and BSD-based distributions).

[0017] The client 110 is in communication with the content server proxy 115 over the client-proxy communication channel 135 and also in communication with the web server 220 over the client-web server communication channel 240. The content server proxy 115 is in communication with the ticket authority 225 over a proxy-authority communication channel 245 and the web server 220 is in communication with the ticket authority 225 over a web server-authority communication channel 250. The content server proxy 115 is also in communication with the content server 120 over a proxy-server communication channel 145. In another embodiment, the web server 220 can communicate with the content server 120 over an agent-server communication channel 255. Similarly, the content server 120 can communicate with the ticket authority 225 over a ticket-content server communication channel 257. In one embodiment, the respective communication channels 135, 145, 240, 245, 250, 255, 257 are established over the network 130.

[0018] In one embodiment, the client 110 includes a web browser 262, such as INTERNET EXPLORER developed by Microsoft Corporation in Redmond, WA, to connect to the web. In a further embodiment, the web browser 262 uses the existing Secure Socket Layer (SSL) support to establish the secure client-web server communication channel 240 to the web server 220. SSL is a secure protocol developed by Netscape Communication Corporation of Mountain View, California, and is now a standard promulgated by the Internet Engineering Task Force (IETF).

[0019] The client 110 may also include an application client 267 for establishing and exchanging communications with the content server 120 over the client-proxy

communication channel 135. In one embodiment, the application client 267 is an ICA client, developed by Citrix Systems, Inc. of Fort Lauderdale, Florida, and is hereafter referred to as ICA client 267. Other embodiments of the application client 267 include an RDP client, developed by Microsoft Corporation of Redmond, Washington, a data entry client in a traditional client/server application, an ActiveX control, or a Java applet. Moreover, the output of an application executing on the content server 120 can be displayed at the client 110 via, for example, the application client 267 or the web browser 262.

[0020] In one embodiment, the content server proxy 115 is a security gateway through which messages over the client-proxy communication channel 135 must pass. In one embodiment, the network firewall 150 repudiates any incoming message from the client-proxy communication channel 135 that does not have the content server proxy 115 as its destination. Likewise, the network firewall 150 repudiates any outgoing message for the client-proxy communication channel 135 unless its source is the content server proxy 115. Although illustrated as a content server proxy 115, the security gateway can alternatively be a router, firewall, relay, or any network component that can provide the necessary security.

[0021] The content server 120 hosts one or more application programs that are available to the client 110. Applications made available to the client 110 for use are referred to as published applications. Examples of such applications include word processing programs such as MICROSOFT WORD and spreadsheet programs such as MICROSOFT EXCEL, both manufactured by Microsoft Corporation of Redmond, Washington, financial reporting programs, customer registration programs, programs providing technical support information, customer database applications, or application set managers.

[0022] In one embodiment, the content server 120 is a video/audio streaming server that can provide streaming audio and/or streaming video to the client 110. In another embodiment, the content server 120 is a file server that can provide any/all file types to the client 110. In further embodiments, the content server 120 can communicate with the client 110 using a presentation protocol such as ICA, from Citrix Systems, Inc. of Ft. Lauderdale, FL or RDP, from Microsoft Corporation of Redmond, Washington.

[0023] In a further embodiment, the content server 120 is a member of a server farm 269, or server network, which is a logical group of one or more servers that are administered as a single entity. In one embodiment, a server farm 269 includes multiple content servers 120, 120′, 120″ (generally 120). Although the embodiment shown in FIG. 2A has three content servers 120, the server farm 269 can have any number of servers. In other embodiments, the server farm 269 is a protected network that is inaccessible by unauthorized individuals, such as corporate Intranet, Virtual Private Network (VPN), or secure extranet. Additionally, the servers making up the server farm 269 may communicate over any of the networks described above (e.g., WAN, LAN) using any of the protocols discussed.

[0024] The ticket authority 225, which in the embodiment shown in FIG. 2A is part of the server farm 269, issues one or more tickets to authenticate the client 110. In particular, the ticket authority 225 enables authentication of the client 110 over one communication channel (i.e., the client-web server communication channel 240) based on authentication credentials. The ticket authority 225 further enables the client 110 to be authenticated to another communication channel (i.e., client-proxy communication channel 135) without having the client 110 repeatedly provide authentication credentials on the other communication channel.

[0025] In one embodiment, the ticket authority 225 is a stand-alone network component. In other embodiments and as shown in FIG. 2B, a modular ticket authority 225, 225′, 225″ is a software module residing on one or more content servers 120. In this embodiment, the web server 220 may communicate with the ticket authority 225 and/or the content server 120 over the agent-server communication channel 255.

[0026] In one embodiment, the ticket authority 225 generates a first ticket and a second ticket. In some embodiments, the tickets are both nonces. In further embodiments, the tickets are generated using a cryptographic random number generator that has been suitably seeded with randomness. The first ticket is transmitted to the client 110 and is used to establish a first communication session between the client 110 and the content server proxy 115. The second ticket is transmitted to the content server proxy 115 and is used to establish a second communication session between the content server proxy 115 and the content server 120.

[0027] The DMZ 230 separates the server farm 269 from the components (e.g., content server proxy 115) of the communications system 205 that are accessible by unauthorized individuals. As described above, the DMZ 230 is delineated with two firewalls 150, 160 that prohibit unauthorized communication. The first firewall 150 and the second firewall 160 each apply a set of policy rules to determine which messages can traverse the DMZ 230. In one embodiment, the first firewall 150 and the second firewall 160 apply the same set of policy rules. Alternatively, the first firewall 150 and the second firewall 160 may apply different sets of policy rules. Each firewall 150, 160 can be a router, computer, or any other network access control device. In another embodiment, the communications systems 205 includes one of the firewalls 150, 160 or no firewall 150, 160.

[0028] In one embodiment, the web server 220 delivers web pages to the client 110. The web server 220 can be any personal computer (e.g., Macintosh computer, a personal computer having an Intel microprocessor, developed by Intel Corporation of Santa Clara, California, a personal computer having an AMD microprocessor, developed by Advanced Micro Devices, Inc. of Sunnyvale, California, etc.), Windows-based terminal, Network Computer, wireless device (e.g., cellular phone), information appliance, RISC Power PC, X-device, workstation, mini computer, main frame computer, personal digital assistant, or other communications device that is capable of establishing the secure client-web server communication channel 240 with the client 110.

[0029] In another embodiment, the web server 220 provides a corporate portal, also referred to as an Enterprise Information Portal, to the client 110. Enterprise portals are company web sites that aggregate, personalize and serve applications, data and content to users, while offering management tools for organizing and using information more efficiently. In other embodiments, the web server 220 provides a web portal, or Internet portal, to the client 110. A web portal is similar to a corporate portal but typically does not include business-specific information.

[0030] The network 130 can be a local-area network (LAN), a wide area network (WAN), or a network of networks such as the Internet or the World Wide Web (i.e., web). The respective communication channels 135, 145, 240, 245, 250, 255, 257 may each be part of different networks. For example, the client-proxy communication channel 135 can belong to a first network (e.g., the World Wide Web) and the client-web server communication channel 240 can belong to a second network (e.g., a secured extranet or Virtual Private Network (VPN)). In other embodiments, the network 130 spans the DMZ 230 as well as the server

farm 269 and the same communication protocol is used throughout. In some embodiments, no firewall 160 separates the content server proxy 115 and web server 220 from the content server 120 and ticket authority 225.

[0031] The client-web server communication channel 240 is any secure communication channel. In some embodiments, communications over channel 240 are encrypted. In certain of these embodiments, the client 110 and the web server 220 may communicate using the Secure Socket Layer (SSL) of the HyperText Transfer Protocol (HTTPS). Alternatively, the client 110 and the web server 220 may use other encryption techniques, such as symmetric encryption techniques, to protect communications.

[0032] Example embodiments of the communication channels 135, 145, 240, 245, 250, 255, 257 include standard telephone lines, LAN or WAN links (e.g., T1, T3, 56kb, X.25), broadband connections (ISDN, Frame Relay, ATM), and wireless connections. The connections over the communication channels 135, 145, 240, 245, 250, 255, 257 can be established using a variety of communication protocols (e.g., HTTP, HTTPS, TCP/IP, IPX, SPX, NetBIOS, Ethernet, RS232, messaging application programming interface (MAPI) protocol, real-time streaming protocol (RTSP), real-time streaming protocol used for user datagram protocol scheme (RTSPU), the Progressive Networks Multimedia (PNM) protocol developed by RealNetworks, Inc. of Seattle, WA, manufacturing message specification (MMS) protocol, and direct asynchronous connections).

[0033] Further, in one embodiment the client-proxy communication channel 135 can be established by using, for example, a presentation services protocol such as Independent Computing Architecture (ICA) protocol, manufactured by Citrix Systems, Inc. of Fort Lauderdale, Florida. ICA is a general-purpose presentation services protocol designed to run

over industry standard network protocols, such as TCP/IP, IPX/SPX, NetBEUI, using industry-standard transport protocols, such as ISDN, frame relay, and asynchronous transfer mode (ATM). The ICA protocol provides for virtual channels, which are session-oriented transmission connections that can be used by application-layer code to issue commands for exchanging data. In other embodiments, the client-proxy communication channel 135 can be established using the thin X protocol or the Remote Display Protocol (RDP), developed by Microsoft Corporation of Redmond, Washington.

[0033] Although described as establishing a first communication session between the client 110 and the content server proxy 115 and a second communication session between the content server proxy 115 and the content server 120, the communication session can be viewed as a single, logical communication session between the client 110 and the content server 120.

[0034] In one embodiment, a user of the client 110 employs the web browser 262 to authenticate the user to the web server 220. In one embodiment, the client 110 transmits user credentials, such as login and password information, to the web server 220. The web server 220 verifies that the user has access to the server network 269.

[0035] In a further embodiment, the web browser 262 uses SSL to establish the secure client-web server communication channel 240. The web browser 262 can alternatively connect to the web server 220 over the client-web server communication channel 240 using other security protocols, such as, but not limited to, Secure Hypertext Transfer Protocol (SHTTP) developed by Terisa Systems of Los Altos, CA, HTTP over SSL (HTTPS), Private Communication Technology (PCT) developed by Microsoft Corporation of Redmond,

Washington, and the Transport Level Security (TLS) standard promulgated by the Internet Engineering Task Force (IETF).

[0036] In one embodiment, the web server 220 transmits a web portal or enterprise portal, as described above, to the client 110 upon validation of the user to enable the client 110 to request an application or a server desktop, for example, to be remotely displayed on the client 110.

[0037] In operation, and also referring to FIG. 3, the client user requests (step 300) content (e.g., an application, a server desktop) to be remotely displayed on the client 110 (i.e., the ICA client 267). In another embodiment, the client 110 uses the web browser 262 to request an application and the web server 220 then authenticates the user. After receiving the request, the web server 220 validates (step 305) the request with the ticket authority 225. The ticket authority 225 then generates (step 310) a ticket, which includes a first ticket, or client ticket, and a second ticket, or content server proxy ticket. The first and second tickets are "one-time use" tickets having no further value after their first use. In a further embodiment, the first and second tickets must be used within a predetermined time period.

[0038] In one embodiment, the ticket authority 225 stores the first and second tickets in memory (e.g., RAM) until the ticket is used. Alternatively, the ticket authority 225 stores the first and second tickets in a storage device (not shown) until the ticket is used. The storage device may include, for example, a database or a persistent memory (e.g., on a floppy disk, hard disk drive). The ticket authority 225 subsequently transmits (step 315) the client ticket to the web server 220 and the web server 220 then forwards (step 320) the client ticket to the client 110.

[0039] The client 110 then initiates (step 325) a communication session with the content server proxy 115 by transmitting a proxy connection request over the client-proxy communication channel 135. The proxy connection request includes the client ticket. In one embodiment, the proxy connection request also includes a dummy password that can be replaced by the content server proxy 115 when establishing a communication session with the content server 120. In a further embodiment, the web server 220 transmits the dummy password to the client 110 for future generation of a proxy connection request having a format acceptable to the content server proxy 115. The content server proxy 115 then extricates (step 330) the client ticket from the proxy connection request and forwards the client ticket to the ticket authority 225 for validation. The ticket authority 225 then validates (step 335) the first ticket. In one embodiment, the ticket authority 225 verifies the first ticket by searching its storage device (e.g., database) for the first expected ticket.

[0040] If the ticket authority 225 does not find the first ticket in the storage device (such as if the first ticket has been used already), the ticket authority 225 ends the communication session. If the received ticket matches the client ticket that the ticket authority 225 expects, the client ticket is validated. The ticket authority 225 then transmits (step 340) the second or content server proxy ticket to the content server proxy 115. Additionally, the ticket authority 225 deletes the client ticket from the storage device, as the client ticket has now been used once. In another embodiment, the ticket authority 225 also transmits the Internet protocol (IP) address of the content server 120 to the content server proxy 115. In yet another embodiment, the ticket authority 225 transmits the domain name of the content server 120 to the content server proxy 115 for future conversion into the IP address.

[0041] The content server proxy 115 receives the second or content server proxy ticket and subsequently opens communications across the proxy-server communication channel 145 by transmitting (step 345) the second ticket to the content server 120. The content server 120 receives the content server proxy ticket and then transmits the ticket over the ticket-content server communication channel 98 to the ticket authority 255 for validation (step 347). In one embodiment, if the ticket authority 225 determines that the content server proxy ticket received from the content server 120 has been used previously or does not have the correct value (i.e., the same value as the value stored in the associated storage device), the ticket authority 225 transmits an error message to the content server proxy 115 (or the web server 220) to terminate the established communication session with the client 110. If the ticket authority 225 validates the content server proxy ticket (step 348), the content server 120 then launches (step 350) the ICA published application. The content server 120 then transmits application information to the content server proxy 115 (step 353) for remote displaying of the application on the client 110 (step 355) using the ICA client 267.

[0042] In a further embodiment, the client 110 launches the ICA client 267 when initiating communications with the content server proxy 115 in step 325. In other embodiments, the client 110 launches the ICA client 267 when the client 110 receives the application information from the content server proxy 115 in step 353.

[0043] Thus, the client 110 is not aware of the content server proxy ticket but only the client ticket. Moreover, the ICA client 267 cannot access the content server 120 without communicating with the content server proxy 115 (and presenting the client ticket).

[0044] The ticket authority 225 could also transmit the content server proxy ticket to the content server proxy 115 in step 340 as the user password for the user of the client 110. This

allows the content server proxy 115 to use the content server proxy ticket as the login password to gain access to the content server 120 without exposing the user's login password over the untrusted part of the web (i.e., the non-secure client-proxy communication channel 135 during step 325). Thus, in one embodiment, the communications system 205 could include a centralized password mapping database managed by the ticket authority 225 and collocated with the content server 120 to map the content server proxy ticket with a user's password.

[0045] Therefore, the password can accompany both tickets (i.e., the content server proxy ticket and the client ticket) or the password can accompany one of the two tickets. As described above, if the password accompanies one of the two tickets, such as the client ticket, then the content server proxy ticket is the password. In one embodiment, the password can be a system password that does not change in value or may be a one-time use password, such as those generated by SecurID tokens developed by RSA Security Inc. of Bedford, Massachusetts.

[0046] Additionally, the invention can be expanded to a communications system having any number of content server proxies 115, or "hops", that the client 110 has to communicate with before establishing a communication session with the content server 120. Although described above and below as a content server proxy 115, a hop can be any network component, such as a firewall, router, and relay.

[0047] For instance and referring to FIG. 4A, a four-hop example is a communication system 405 having a first content server proxy 115′, a second content server proxy 115′′, and a third content server proxy 115′′′ (generally 115). The content server proxies 115 communicate over a proxy-proxy communication channel, such as a first proxy-proxy

communication channel 410′ and a second proxy-proxy communication channel 410′′ (generally proxy-proxy communication channel 410). The client 110 communicates with the first content server proxy 115′ which communicates with the second content server proxy 115′′. The second content server proxy 115′′ communicates with the third content server proxy 115′′′ and then the third content server proxy 115′′′ communicates with the content server 120 over the proxy-server communication channel 145 to establish the communication session with the content server 120. Further, although the embodiment described above includes a ticket having a client ticket and a content server proxy ticket, another embodiment includes the ticket comprising numerous tickets.

[0048] More explicitly and also referring to FIG. 4B, the web server 220 receives a request from the client 110 for an application and the web server 220 validates the request with the ticket authority 225 (step 405). The ticket authority 225 then generates an N part ticket (e.g., T_1 to T_N) in step 410. In one embodiment, the ticket authority 225 then transmits a portion T_i of the N part ticket (e.g., the first part of the ticket, or first ticket T_1) to the web server 220 (step 415). The web server 220 then transmits the ticket T_1 to the client 110 (step 420). In one embodiment, the ticket authority 225 also transmits the address of the next "hop" (e.g., the first content server proxy 115′) to the web server 220, which then transmits the address to the client 110. This address is the address of the next hop (e.g., content server proxy 115) that this hop (e.g., client 110) needs to communicate with for the client 110 to eventually be authenticated to the content server 120.

[0049] The client 110 uses the address to then contact the next "hop" (e.g., first content server proxy 115′) and initiates a communication session with the first content server proxy

115′ by transmitting a proxy connection request over the client-proxy communication channel 135. The first content server proxy 115′ then extracts (step 430) the first ticket T_1 from the proxy connection request and forwards this ticket to the ticket authority 225 for validation. The ticket authority 225 then validates (step 435) the first ticket T_1.

[0050] Upon proper verification of the first ticket T_1, the ticket authority 225 transmits the next ticket T_i from the N part ticket (e.g., T_2) to the next content server proxy 115 (e.g., first content server proxy 115′) (step 440). In some embodiments, the ticket authority 225 also transmits the address of the next hop (e.g., the second content server proxy 115′′) to this hop (e.g., the first content server proxy 115′). The first content server proxy 115′ transmits this ticket to the next hop (e.g., the second content server proxy 115′′) (step 445). In one embodiment, the second content server proxy 115′′ verifies T_2 by transmitting the ticket to the ticket authority 225 (step 450). The ticket authority 225 validates the second ticket T_2 (step 455) and the process continues, as shown in steps 460 through 475. Once the last part of the N part ticket has been validated, steps 350 through 355 occur, as shown in FIG. 3, to launch the application on the client 110.

[0051] In one embodiment, each content server proxy 115 (i.e., each hop) validates T_i (e.g., T_2) with a ticket authority 225 associated with the content server proxy 115 (i.e., hop). In this embodiment, after each content server proxy 115 validates the ticket T_i (e.g., T_2) with a ticket authority 225, the ticket authority 225 at which the validation took place transmits the next ticket T_{i+1} (e.g., T_3) and the address of the next content server proxy 115 (i.e., the next "hop" destination) to the content server proxy 115 that had validated the ticket T_i. Thus, each content server proxy 115 is associated with a ticket authority 225 that has been configured with the current and next hop tickets (i.e., validating T_i and transmitting T_{i+1} for

the next hop). Consequently, the next content server proxy 115 acts as the client for that hop. This process is repeated until reaching the content server 120 in the communications system 405. Thus, each hop has been validated individually without revealing all of the ticket to any one hop.

[0052] In other embodiments, the ticket authority 225 may issue more than one ticket rather than issuing one ticket having many parts. For example, the ticket authority 225 generates a first hop ticket and a second hop ticket in step 410, where the first hop ticket has no association with the second hop ticket. The ticket authority 225 subsequently transmits the first hop ticket to the web server 220 and the web server 220 transmits the first hop ticket to the client 110. The client 110 transmits this first hop ticket to the content server proxy 115 (e.g., first content server proxy 115′) for validation by the ticket authority 225. Upon validation in step 435, the ticket authority 225 transmits in step 440 the second hop ticket to the next content server proxy 115 (e.g., second content server proxy 115′′) while the first hop ticket is independent from the second hop ticket.

[0053] In a further embodiment, one or more of the ticket authorities 225 provides the content server proxies 115 with any necessary information needed to connect to the next hop, such as, but without limitation, encryption keys, SSL method configuration information, and authentication information to connect to a SOCKS server (e.g., SOCKS5 server, developed by NEC Corporation of Tokyo, Japan).

[0054] In yet another embodiment, a ticket authority 225 only generates a single ticket. The ticket authority 225 transmits the single ticket to the web server 220. The web server 220 forwards the single ticket to the client 110. The content server proxy 115 subsequently receives the ticket from the client 110 and "consumes" the single ticket upon validation. As

a result, the communications system 205 can use a single ticket to provide the ability to use arbitrary communication protocols over the client-proxy communication channel 135 and the client-web server communication channel 240. Additionally, because the content server 120 does not receive or verify the single ticket, the ticket is transparent to the content server 120 and, consequently, the content server 120 is not "aware" of the use of the ticket.

[0055] By exploiting the security of the secure communications between the client 110 and the web server 220 over the secure client-web server communication channel 240, the communications system 205 establishes a secure communication link over the non-secure client-proxy communication channel 135 to remotely display desktop applications securely on the client 110.

[0056] In yet another embodiment and referring again to FIG. 3, the ticket authority 225 transmits in step 315 a disabled version of the content server proxy ticket with the client ticket to the web server 220 for transmission to the client 110. The client 110 subsequently transmits (step 325) the content server proxy ticket along with the client ticket to the content server proxy 115 as part of the proxy connection request. The content server proxy 115 then forwards both tickets to the ticket authority 225. Upon receiving a disabled content server proxy ticket, the ticket authority 225 enables the content server proxy ticket after validating the client ticket. The ticket authority 225 then transmits the enabled content server proxy ticket to the content server proxy 115 for authentication to the content server 120.

[0057] Alternatively, in another embodiment the web server 220 receives a disabled content server proxy ticket and an enabled client ticket from the ticket authority 225 and only transmits the client ticket to the client 110. The client 110 transmits (step 325) the client ticket to the content server proxy 115 as part of the proxy connection request. The content

server proxy 115 then forwards the client ticket to the ticket authority 225. The ticket authority 225 validates the client ticket and, upon validation, enables the content server proxy ticket previously transmitted to the web server 220. In yet another embodiment, the ticket authority 225 transmits an enabled content server proxy ticket to the web server 220 upon validation of the client ticket for authentication to the content server 120.

[0058] Thus, at any given time, the ticket authority 225 provides only one ticket that is enabled to the client 110 or content server proxy 115 that the ticket authority 225 can validate. The ticket authority 225 may provide another ticket that can't be validated (i.e., a disabled ticket) until the enabled ticket is validated. Alternatively, the ticket authority 225 may not transmit the content server proxy ticket to the content server proxy 115 until the ticket authority 225 validates the enabled ticket. As discussed in further detail below, this enforces network routing of communications using the communications system 205 because the client 110 cannot traverse the web server 220 or the content server proxy 115 without having the ticket authority 225 validate the enabled ticket and transmit the ticket needed to communicate with the content server 120.

[0059] In another embodiment, instead of transmitting the content server proxy ticket to the content server proxy 115 as in step 340, the ticket authority 225 transmits the content server proxy ticket to the web server 220 directly over the web server-authority communication channel 250. The web server 220 then automatically transmits the content server proxy ticket to the content server 120. In other words, the web server 220 "pushes" the content server proxy ticket to the content server 120. The ticket authority 225 can also push the content server proxy ticket to the content server 120 without transmission of the content server proxy ticket to the content server proxy 115 or the web server 220.

[0060] In yet another embodiment, the content server 120 retrieves the content server proxy ticket from the ticket authority 225 over the ticket-content server communication channel 257. In other words, the content server 120 "pulls" the content server proxy ticket from the ticket authority 225. The above examples are illustrations of techniques used to eliminate step 345 (while modifying the destination of the transmission in step 340).

[0061] Moreover, the invention enforces the routing of the client 110 through the content server proxy 115. As stated above, the client 110 has to possess the content server proxy ticket to establish a communication session with the content server 120. More specifically, to establish a connection with the content server 120, the web server 220 first has to validate the request of the client 110 with the ticket authority 225. Once validated, the client 110 obtains the first ticket and transmit this first ticket to the ticket authority 225 for validation. However, upon validation, the ticket authority 225 transmits the content server proxy ticket back to the content server proxy 115 rather than the client 110. The communication session between the client 110 and the content server 130 is established when the content server 130 receives the content server proxy ticket. Thus, the client 110 has to communicate with the content server proxy 115 in order to have the content server proxy ticket transmitted to the content server 130, thereby enforcing the routing of the client 110 through the content server proxy 115. Thus, the invention can ensure the proper traversal of a security device (e.g., the content server proxy 115) before granting access to the content server 120.

[0062] For example, a content server 120 executes several applications, such as MICROSOFT WORD and MICROSOFT EXCEL, both developed by Microsoft Corporation of Redmond, Washington. In one embodiment, the client 110 uses NFUSE, developed by Citrix Systems, Inc. of Fort Lauderdale, Florida, to obtain information from the server farm

269 on which applications can be accessed by the client 110. If a client user wants to access and use MICROSOFT WORD, the client 110 requests the application from the web server 220. However, only users who pay an application fee for MICROSOFT WORD can become authorized to access the application.

[0063] To ensure the payment of the application fee, the communications system 205 includes the content server proxy 115 and the ticket authority 225 to enforce the routing of the client 110 through the content server proxy 115. The routing of the client 110 through the content server proxy 115 is valuable to the application provider if the content server proxy 115 is used to collect the application fee and authorize the user for access to the application.

[0064] The ticket authority 225 subsequently generates a ticket associated with the request for the application. An enabled first ticket is then transmitted to the client 110. Because the client 110 does not have the address of the content server 120, the client 110 cannot access the application. Further, the client 110 has not been authorized by the content server proxy 115 yet (i.e., has not yet paid). Thus, the client 110 has to communicate with the content server proxy 115 to become authorized. The content server proxy 115 can then transmit the enabled first ticket to the ticket authority 225 upon payment of the application fee.

[0065] The ticket authority then validates the client ticket and subsequently transmits (or enables) a content server proxy ticket to the proxy 115. The content server proxy 115 then transmits the content server proxy ticket to the content server 120 (e.g., assuming the client user has paid the application fee), which enables the content server 120 to transmit the application to the client 110. The communications system 205 may also use Application Launching And Embedding (ALE) technology, developed by Citrix Systems, Inc., to enable

the launching of the application from or the embedding of the application into an HTML page for delivery to the client 110.

[0066] Having described certain embodiments of the invention, it will now become apparent to one of skill in the art that other embodiments incorporating the concepts of the invention may be used. Therefore, the invention should not be limited to certain embodiments, but rather should be limited only by the spirit and scope of the following claims.